

//build/

# TouchDevelop:

A touch-first IDE for the Web created with  
TypeScript

Michał Moskal  
Researcher  
3-018

//build/

```
***** COMMODORE 64 BASIC V2 *****  
64K RAM SYSTEM 38911 BASIC BYTES FREE  
READY.  
10 PRINT "HELLO WORLD"  
20 GOTO 10  
RUN
```

This was my first computer.  
It wanted to be programmed.

# These are their first computers



Hi-res displays, hi-powered CPUs, fancy sensors, network.

No keyboard to speak of.

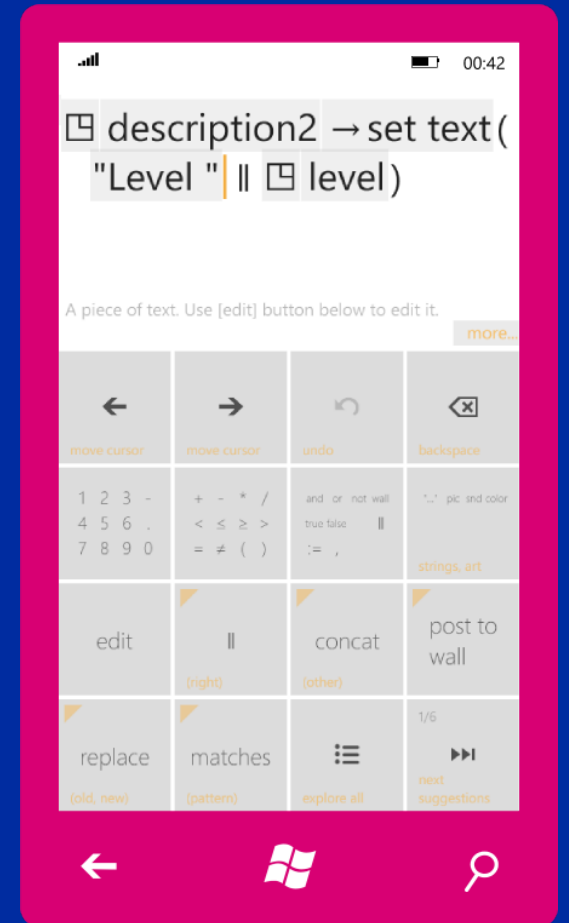
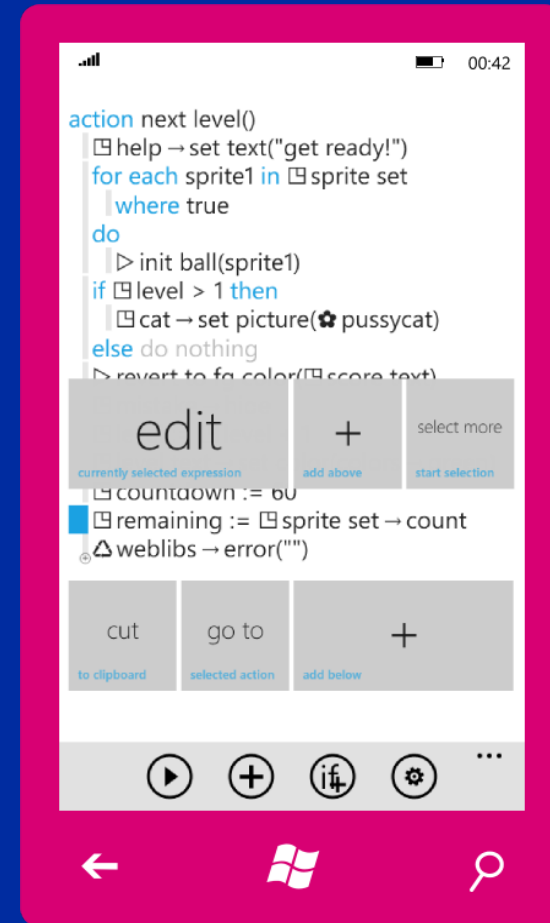
No BASIC prompt.

At Microsoft Research we try to see what's possible.

Can we program  
on these things?

# Programming the phone – no PC required!

TouchDevelop is a Windows Phone 7 app bringing the joy of programming 8-bit computers to the smartphone age.



Demo: TouchDevelop on the phone

# It actually works!

200,000+  
downloads

Full social  
experience

10,000 scripts

Great for education



<http://touchdevelop.com/gallery/>

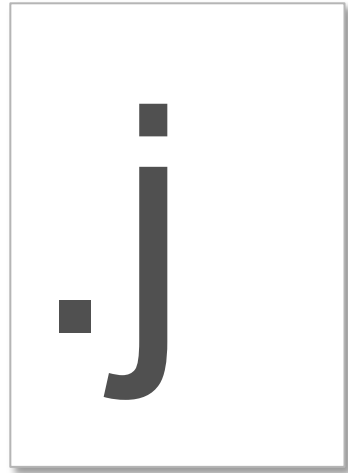


Bigger is better.  
More is more.  
Make it a webapp!



What's a web  
app?

**HTML**



**S**



What do we  
have?

152,000 lines of C#

12,000 lines of  
XAML

125,000 lines of C#

(plus 10,000 lines of tools in C#)

# HTML5 & CSS3 instead of XAML



It's immediate (just refresh the page)

It's popular (just bing around for answers)

It's fast (CSS animations, rendering speed)

It's simple (e.g., list viewer is just a

# Now to that JavaScript thing...

.js

It's immediate (just refresh the page)

It's popular (just bing around for answers)

It's **reasonably** fast (with modern JITs)

It's **not** simple (e.g., 20 ways of doing OO)

# I really like types

TypeScript makes for an easier C# to JavaScript transition.

TypeScript – a superset of JavaScript, compiles to idiomatic JS and adds:

- Type annotations and inference

- Syntax for classes, interfaces & modules

- Enables tool support (in VS and web)

- In developer preview

# TypeScript $\supseteq$ JavaScript

What goes in JavaScript goes in TypeScript.

For better or worse.

You can skip type annotations or just use `:any`

You can still do crazy JS stuff

Even `eval()` is still there and reflection is super-easy. Great for interpreters/compilers ;-)

You can copy&paste JS snippets from the web

You can use existing JS libraries

Type annotations for DOM, WinRT and jQuery are included – you get type-checking there!

# C# --> TypeScript (classes)

```
namespace Microsoft.TouchDevelop.Runtime {  
    public sealed class SpriteRepr  
        : ReferenceRepr  
    {  
        private GameBoardRepr parent;  
        private List<SpringRepr> springs  
            = new List<SpringRepr>();  
  
        public Vector2 ComputeForces(  
            Vector4 posSpeed)  
        { ... }  
    }  
}
```

```
module TDev.RT {  
    export class Sprite  
        extends RTValue  
    {  
        private parent: Board;  
        private springs: Spring[] = [];  
  
        public computeForces(  
            posSpeed: Vector4): Vector2  
        { ... }  
    }  
}
```

# C# --> TypeScript (statements)

```
foreach (var spring in this.springs) {  
    force +=  
        spring.ForceOn(this);  
}
```

```
for (int i = 0; i < touchPts.Count;  
    i++)  
{  
    var unitNormal = touchPts[i];  
    var d =  
        Vector2.Dot(unitNormal, force);  
    if (d > 0) continue;  
    force = unitParallel * d;  
}
```

```
this.springs.forEach((spring) => {  
    force =  
        force.add(spring.forceOn(this));  
}))
```

```
for (var i = 0; i < this.touchPts.length;  
    i++)  
{  
    var unitNormal = this.touchPts[i];  
    var d =  
        Vector2.dot(unitNormal, force);  
    if (d > 0) continue;  
    force = unitParallel * d;  
}
```



Demo: TypeScript premiere

# Define a class

```
class Greeter {
    greeting: string;
    constructor (message: string) {
        this.greeting = message;
    }
    greet() {
        echo("Hello, " + this.greeting);
    }
}

var greeter = new Greeter("BUILD");
greeter.greet();
```

# Override a method, with JS flavor

```
class Greeter {  
    constructor (public greeting: string) {}  
    greet(s:string) {  
        echo(s + " " + this.greeting)  
    }  
}
```

```
Greeter.prototype.greet = function (s) {  
    echo(s + " and " + this.greeting);  
};
```

# I was always missing unlink() in HTMLElement

```
interface HTMLElement {  
    unlink():void;  
}
```

```
HTMLElement.prototype.unlink = function () {  
    if (this.parentNode)  
        this.parentNode.removeChild(this);  
}
```

# Interfaces are for classes, right?

```
module Cloud1 {  
    export function fetch(path:string) { return ""; }  
    export function somethingElse() { ... }  
}  
  
interface CloudIface {  
    fetch(path:string):string;  
}  
  
var Cloud:CloudIface = Cloud1;  
  
Cloud1.somethingElse = () => { };
```

# The 'this' pointer – you *\*will\** get confused

```
class Greeter {
  constructor (public greeting: string) {}
  greet() {
    // the difference between function and =>
    [2011, 2012].forEach(function (n) {
      echo(this.greeting + " " + n);
    })
  }
}
```

# There is no block scope

```
function echo(s) => console.log(s);
var fruits = ["pears", "oranges", "bananas"];
var callbacks = [];
var i;
for (i = 0; i < fruits.length; ++i) {
  callbacks.push(() => echo(fruits[i]))
}
callbacks.forEach((cb) => cb());
// prints: undefined, undefined, undefined
// i == fruits.length here, which is out of range

callbacks = [];
for (i = 0; i < fruits.length; ++i) {
  var fruit = fruits[i];
  callbacks.push(() => echo(fruit))
}
callbacks.forEach((cb) => cb());
// prints: bananas, bananas, bananas
// there is just one copy of 'fruit' variable; the loop/block doesn't introduce a scope!

(() => {
  var fruit = fruits[i];
  callbacks.push(() => echo(fruit))
})();
callbacks.forEach((cb) => cb());
// prints: pears, oranges, bananas

// if you really need a scope inside of a loop - create a function and call it immediately:
callbacks = [];
for (i = 0; i < fruits.length; ++i) {
```

# It's a functional language

```
callbacks = [];  
fruits.forEach((fruit) => {  
    callbacks.push(() => echo(fruit))  
})  
callbacks.forEach((cb) => cb());  
// prints: pears, oranges, bananas  
// as expected; it's also shorter
```

```
// speaking of short - this also  
works:  
// (or should; right now it seems to  
crash the compiler)  
callbacks = fruits.map((fruit) =>  
{ return () => echo(fruit) })  
callbacks.forEach((cb) => cb());  
// prints: pears, oranges, bananas
```



# Don't use for(each)

```
// no no no
```

```
for (var v in obj) {  
    ...  
}
```

```
// yes yes yes
```

```
Object.keys(obj).forEach((v) => {  
    ...  
})
```

# TypeScript highlights

The high-order bit.

Smoother migration from C# to crazy JavaScript.

Avoids lots of runtime errors  
(even though there are no 100% guarantees)

IntelliSense is useful

Usually quite functional style

A green trapezoidal graphic on the left side of the slide, tapering from top to bottom. It contains the text "The outcome" in white.

## The outcome

WP7 App:

152,000 lines of C#

12,000 lines of XAML

Web App:

47,000 lines of TypeScript

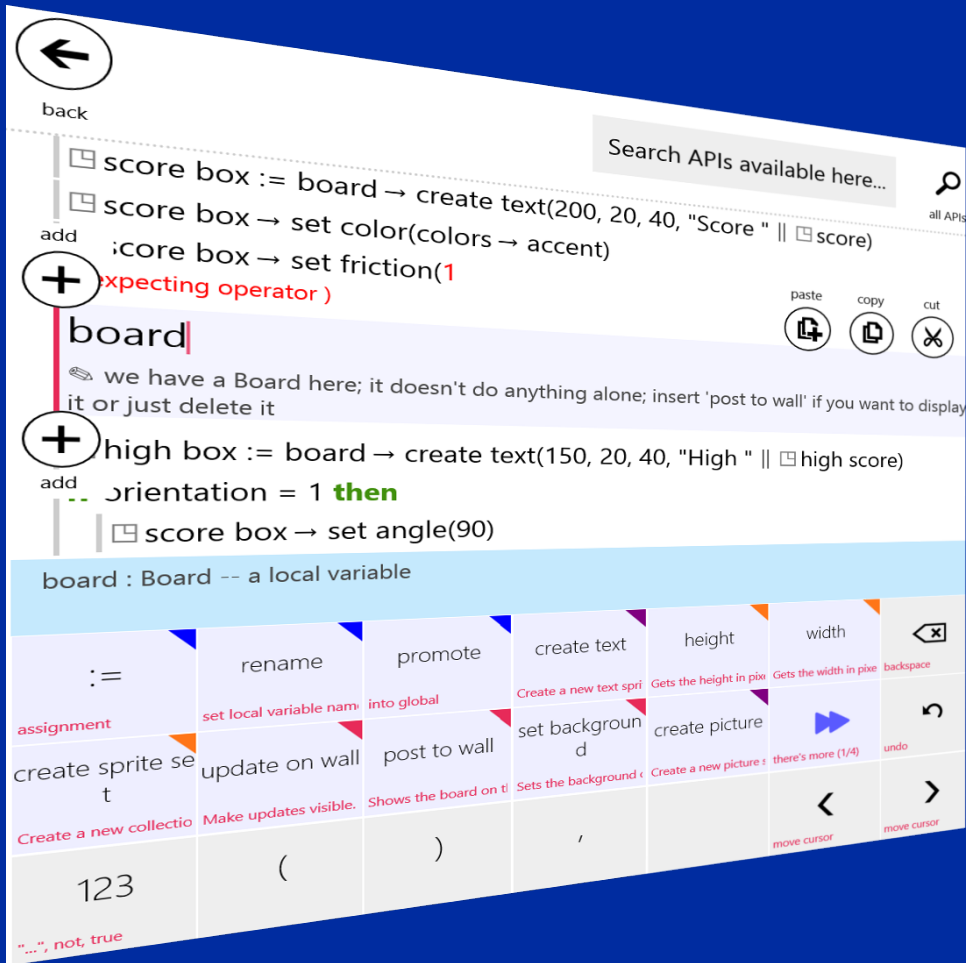
3,500 lines of CSS

79 lines of HTML

(but we're not quite done yet)

# Demo: TouchDevelop on the Web

# Take aways

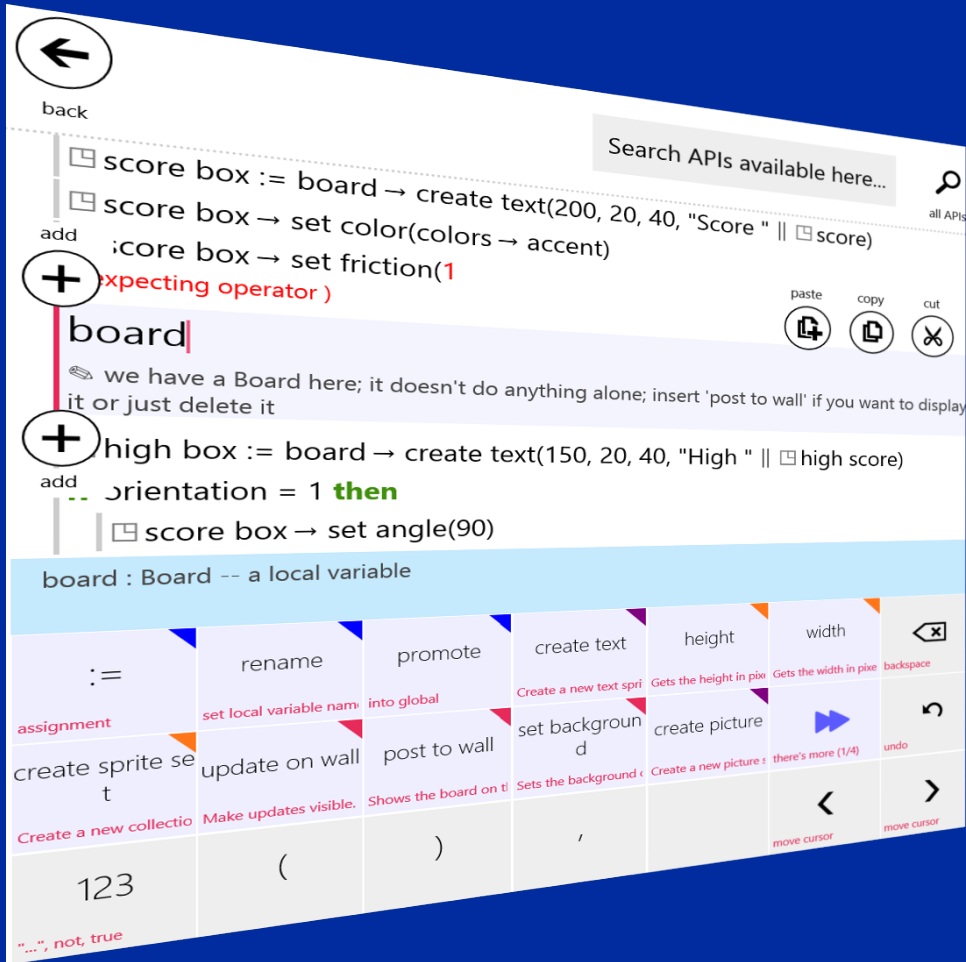


Big apps? TypeScript is going to help you!

Types are good.

TouchDevelop is cool :-)

# Resources



[touchdevelop.com/  
app](http://touchdevelop.com/app)

[typescriptlang.org](http://typescriptlang.org)

# Questions?



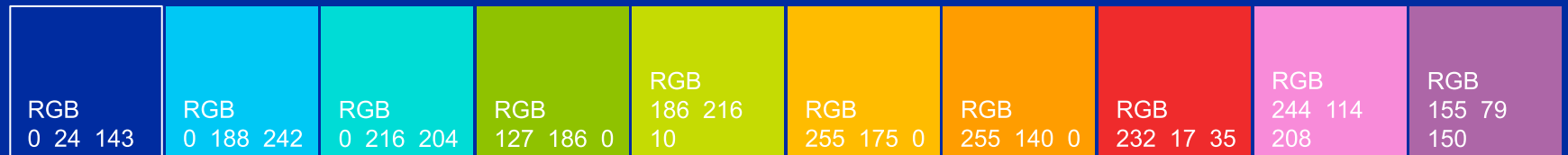
# Color palette

Lead and accent colors have been formatted into this template. In general, use these 4 colors for all slides.

//build/ template colors



All Windows brand colors



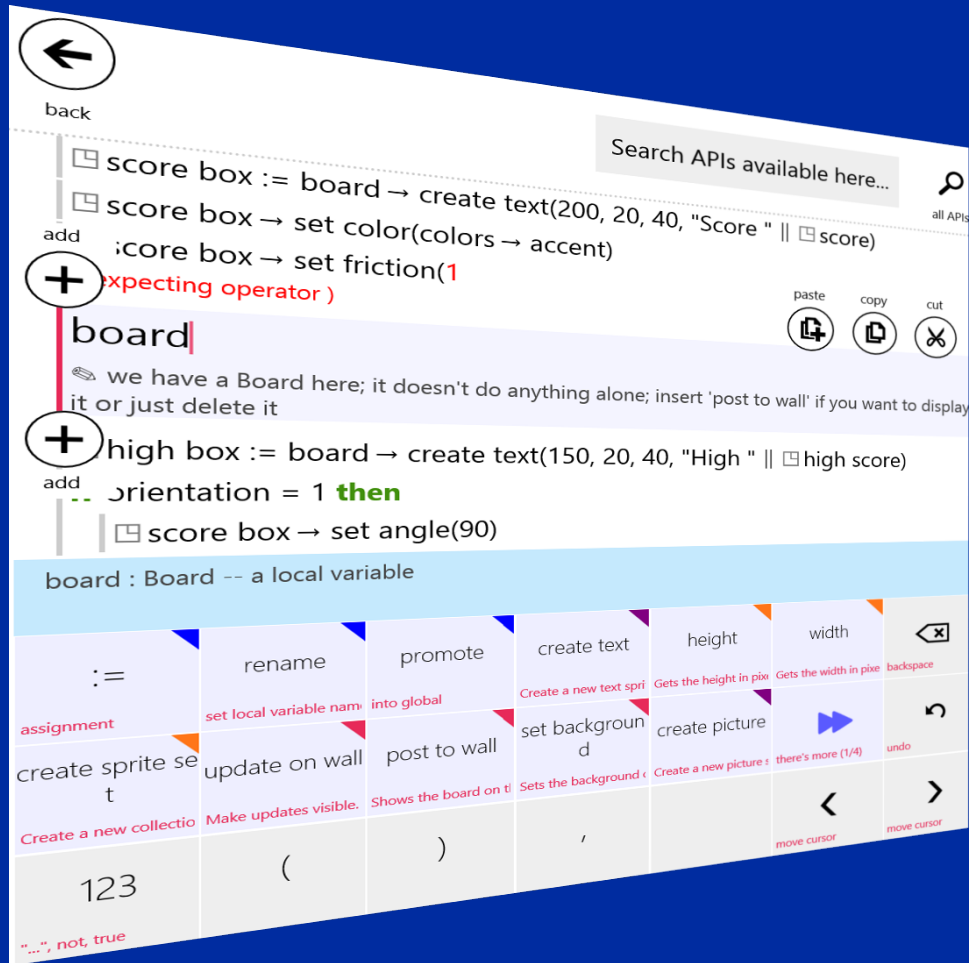


# Table

This is a content  
box, always  
24pts and  
sentence case

Component	C#	XAML	TypeScript	CSS+HTML
Cloud	125,000			
Client	152,000	12,000	47,000	3,500
Tools	10,000	300		

# TouchDevelop is a complex webapp



Not a replacement for professional development tools.

But it does involve a compiler, intellisense engine, lots of UI code for the editor, etc.

# Things I didn't know existed

The web development has changed in the last couple years

# Client-side storage

## localStorage:

easy, sync, cross-browser, 5MB max

```
localStorage["key"] = "value";
```

## IndexedDB & WebSQL

async, often 50MB max

IndexedDB – W3C, IE, Firefox

WebSQL – Webkit (Safari,  
Chrome)

http://localhost/#edit:e25b05ca-555f-4031-6309-fcd3de2df558:setupsc

missile defense (event base... x

Search code...

- ▷ distance squared  
(sprite(), x, y) : Number : an action
- ▷ draw city  
(width, maxHeight) : Picture : an action
- ▷ draw landscape  
(pic, maxHeight) : Number Map : an action
- ▷ game over(board)  
an action
- ▷ play wave(wave1)  
an action
- ▷ reset high()  
an action
- ▷ setup scores  
(board, orientation) : an action
- ▷ still blowing up

board

- we have a Board here; it doesn't do anything alone; insert 'post to wall' if you want to display it or just delete it
- high box := board → create text(150, 20, 40, "High " || high score)
- if orientation = 1 then
  - score box → set angle(90)
  - score box → set pos(board → width, 100)
  - high box → set angle(90)
  - high box → set pos(board → width, board → height - 150)
- else do nothing
- high box → set friction(1)
- if orientation = 3 then
  - score box → set angle(- 90)
  - score box → set pos(0, board → height - 100)
  - high box → set angle(- 90)
  - high box → set pos(0, 150)

File Find Disable View Images Cache Tools Validate | Browser Mode: IE10 Document Mode: Standards

HTML CSS Console Script Profiler Network Search HTML...

```

<div class="sideTabScroll tabNav" style="overflow-y: scroll; box-sizing: border-bo
  <div class="navTopSpacer" style="height: 1em;" data-errors="no" data-selected="n
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <div class="navHeader" data-errors="no" data-selected="no">
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <div class="navItemInner">
      <div class="navImg" style="background-color: rgb(231, 42, 89);">
        <div class="navContent">
          <div class="navName">
            <div class="navDescription">
          </div>
        </div>
      </div>
    </div>
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-errors="no" data-selected="no" data-c
    <button class="navItem" type="button" data-active="no" data-errors="yes" data-sa
  </div>

```

Style Trace Styles Layout Attributes

- inherited - body
  - body
    - color: black;
    - font-family: "Segoe UI", Sans-Serif;
    - font-size: 24px;
  - inline style
    - font-size: 16px;
- inherited - button
  - button, input[type=button], .stmt
    - cursor: pointer;
  - .navItem
    - text-align: left;
    - font-family: inherit;
    - font-size: 1em;
- inherited - div
  - .navItemInner
    - font-size: 0.7em;
  - .navItemInner, .navHeader
    - font-family: "Segoe UI Light", "Segoe UI", Sans-Serif;
    - font-weight: 100;

```

34887     };
34888     Editor.prototype.recompileScript = function () {
34889         var cs = TDev.AST.Compiler.getCompiledScript(TDev.Script, false, this.withTracing, this);
34890         this.currentRt.initFrom(cs);
34891     };
34892     Editor.prototype.runAction = function (a, args) {
34893         if (typeof args === "undefined") { args = null; }
34894         var _this = this;
34895         this.recompileScript();
34896         var missing = this.currentRt.compiled.missingApis;
34897         if(!this.complainedAboutMissingAPIs && missing.length > 0) {
34898             TDev.ModalDialog.ask("the following APIs are not implemented on the current device:");
34899             this.complainedAboutMissingAPIs = true;
34900             _this.runAction(a, args);
34901         });
34902         return;
34903     }
34904     var rt = this.currentRt;
34905     rt.setHost(this.host);
34906     this.host.showWall();
34907     this.initPageStack();
34908     TDev.SizeMgr.applySizes();
34909     if(!(a instanceof TDev.AST.Action)) {
34910         a = null;
34911     }
34912     if(a == null || !(a.isRunnable())) {
34913         rt.postText("can't run this");
34914     } else {
34915         var isMain = a == TDev.Script.mainAction();
34916         function setHash() {
34917             if(isMain) {
34918                 _this.historyMgr.setHash("run:" + TDev.Script.localGuid, TDev.Script.getName(a));
34919             } else {
34920                 _this.historyMgr.setHash("run-action:" + TDev.Script.localGuid + ":" + a.stableName);
34921             }
34922         }
34923         setHash();
34924         var name = a.stableName;
34925         function runIt() {
34926             TDev.Ticker.tick(TDev.Ticks.coreRun, _this.currentPublicId);
34927             var fn = rt.compiled.actionsByStableName[name];
34928             rt.run(fn, args);
34929         }
34930         this.runAction = function () {
34931             TDev.tick(TDev.Ticks.coreRerun);

```

Name	Value	Type
this	{...}	Object, (Editor)
a	{...}	Object, (Action)
args	null	Null
arguments	{...}	Object, (Arguments)
[Methods]	{...}	
[prototype]	{...}	Object
[0]	{...}	Object, (Action)
length	1	Number
setHash	function setHash() { ... }	Object, (Function)
runIt	function runIt() { ... }	Object, (Function)
this	{...}	Object, (Editor)
[prototype]	{...}	Object, (Editor)
actionProperties	{...}	Object, (ActionProperties)
actionView	{...}	Object, (ActionView)
auxRenderer	{...}	Object, (Renderer)
calculator	{...}	Object, (Calculator)
clipMgr	{...}	Object, (ClipMgr)
codeInner	{...}	[Object, HTMLDivElement]
codeOuter	{...}	[Object, HTMLDivElement]
codeViews	[[object Object],[object Obj...]	Object, (Array)
commentEditor	{...}	Object, (CommentEditor)
complainedAb...	false	Boolean
currentCodeVi...	{...}	Object, (ActionView)
currentPublicId	""	String
currentRt	{...}	Object, (Runtime)
currentSideTab	{...}	Object, (ScriptNav)
currentStmntEd...	null	Null
doingRefresh	false	Boolean
emptyScriptText	"meta version 'v2.2';\nmeta ..."	String
fieldEditor	{...}	Object, (FieldEditor)
forceRefresh	false	Boolean
forceVersionC...	false	Boolean
historyMgr	{...}	Object, (HistoryMgr)
host	{...}	Object, (EditorHost)
keyMgr	{...}	Object, (KeyboardMgr)
landscapeSea...	{...}	[Object, HTMLDivElement]
lastDed	{...}	Object, (Action)
lastSideTab	{...}	Object, (ScriptNav)
lastTappedNode	{...}	Object, (ExprStmnt)
lastTapTime	1350326900334	Number
libCache	{...}	Object, (LibraryCache)
librefProperties	{...}	Object, (LibraryRefProperties)
loadDedImme...	""	String
paneState	0	Number

# CSS3 transitions and animations

```
@keyframes showDn {  
  0% {  
    opacity: 0;  
    transform:  
      translate(0, 2em);  
  }  
  100% {  
    opacity: 1;  
    transform: none;  
  }  
}  
  
.show {  
  animation:  
    showDn 0.3s ease-out;  
}
```

And browser  
prefixes :(

```
@-webkit-keyframes showDn {  
  0% {  
    opacity: 0;  
    -webkit-transform:  
      translate(0, 2em);  
  }  
  100% {  
    opacity: 1;  
    -webkit-transform: none;  
  }  
}  
  
.show {  
  -webkit-animation:  
    showDn 0.3s ease-out;  
}
```



# CSS3 transitions and data-\*

```
<button data-active="no">  
    Click here!  
</button>  
  
.button[data-active^="yes"] {  
    font-size: 1.5em;  
    transition: font-size 0.5s;  
}
```

# Promises

```
readLineAsync()  
  .then((s:string) =>  
    db.getRecord(s))  
  .then((r:MyRecord) => {  
    ui.showDialog(r.foobar)  
  }).done();
```

```
Promise.join({ db: getDbAsync(),  
              id: userIdAsync() })  
  .then((ctx) =>  
    ctx.db.getAsync(ctx.id))
```




# Pains of single-origin policy

By default www.foo.com can only make HTTP requests to www.foo.com.

Great for cookies, horrible for general programming.

Webserver can allow other domains via CORS.



We didn't yet  
get there...

WebSockets: let you keep  
telnet-like connection open.

WebWorkers: let you run stuff  
in background and  
communicate via message-  
and resource-passing

# TypeScript wish list

Nobody's  
perfect.

Direct debugger support

Faster IntelliSense

Generics

Stricter strict mode

“var” scoping, “this” typing, stricter “any”

Conditional compilation

Algebraic data-types



<http://touchdevelop.com/app/>

Bigger screens.

More devices.

Take it to the

Web!

